AD-A261 398

NTATION PAGE

(2)

FINAL 01 NOV 91 TO 31 OCT 92

4 TITLE AND SUBTITLE
LINEAR-PROGRAMMING TOOLS IN INTEGER PROGRAMMING:
THE TRAVLEING SALESMAN

2304/DS
F49620 92 J 0053

6 AUTHOR(S)
DR. ROBERT BIXBY

7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RICE UNIVERSITY
DEPARTMENT OF MATHEMATICAL SCIENCES
P.O. BOX 1892
HOUSTON TX 77251-1892

AFOSR-TR- 2

9 SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 DUNCAN AVE, SUTE B115
BOLLING AFB DC 20332-0001

DTIC
ELECTE
MAR 05 1993
S E D

10 SPONSORING MONITORING AGENCY REPORT NUMBER
F49620-92-J-0053

93-04646

11 SUPPLEMENTARY NOTES

12a DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

12b DISTRIBUTION CODE

13 ABSTRACT (Maximum 200 words)

New branching rules and new methods to control the size of LP's resulted in a world's record for the solution of large traveling salesman problems.

14 SUBJECT TERMS

15 NUMBER OF PAGES

16 PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT	18 SECURITY CLASSIFICATION OF THIS PAGE	19 SECURITY CLASSIFICATION OF ABSTRACT	20 LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

NSN 7540-01-280-5500

Standard Form 298 Rev
Prescribed by ANSI Std
298-102

93 3 4 022

Final Report on AFOSR Grant #F49620-92-J-0053

Investigations in Linear and Integer Programming

Principal Investigator: Robert E. Bixby, Rice University

Linear-Programming Tools in Integer Programming: The Traveling Salesman Problem

A new "world record" was set in the traveling salesman problem. This work, joint with D. Applegate, V. Chvátal and W. Cook, was designated by *Discover Magazine* as one of its top ten science stories of the year. The AFOSR support of work on linear-programming tools for integer programming provided partial support for this TSP project.

The problem we solved had 3038 nodes. The total computation time was equivalent to approximately one and one-half years of SPARC 2 seconds. The previous "record" was 2392 nodes, a problem that we can now solve in under two hours on a SPARC 2 (with no branching!). In addition to solving the 3038 problem, we solved 15 additional previously unsolved real-world TSP instances (mostly VLSI drilling problems).

We are now in the process of preparing to write up the results of the above work, a project that we hope will be completed within the next six months. From the point of view of integer programming, the impact of this work should go well beyond the TSP. Several new separation techniques were developed along with new methods for controlling the sizes of the the linear programs that result in any branch-and-cut approach. Other consequences were improved general-purpose branching rules and improvements in the interaction of the LP-solver with the branching code (not to mention an almost two-orders-of-magnitude improvement in the LP solutions times for problems in this class).

Basis Recovery

A joint paper with Matt Saltzman, "Recovering an optimal LP basis from an interior point solution," was completed and submitted for publication. In this paper we consider a primal version of the *basis recovery* problem for interior point methods: Given an optimal primal solution x^* for some linear program, construct an optimal basic solution B^*. The essentials of that procedure are:

- Construct some "numerically stable" starting basis B that includes as many "positive" elements from x^* as possible.
- After the construction of the initial B, there will, in general, remain some "positive" components of x^* that are outside of B. Such variables are designated as *superbasic*. The second part of the procedure is to modify the simplex method to deal with these superbasic variables.

Results from this approach were far superior to those for any previous basis-recovery procedure. In tests on the Netlib problem set, this method was able to recover an optimal basis in 5% of the interior-point solution time for most problems.

Since the completion of this work there have been further significant developments. In particular, a procedure proposed by Megiddo has been investigated and shown to be very promising for problems that are highly primal degenerate. This procedure has a time-complexity that is linear in the total number of rows and columns in the given linear program.

Basis Reduction for Integer Programming

The object of this research was to study a method that is particularly well suited for mixed integer programming problems with a relatively small number of general (non 0/1) integer variables. The drawback of this method has been that the computational effort grows quickly with the number of integer variables in the problem being solved. Thus, with current technology, the number of integer variables is essentially restricted to 100 or fewer. The aim of our research has been to extend the applicability of this method by studying how it can be applied in parallel. In the process, we have not simply been aiming to implement a known algorithm in a new computing environment,

but to investigate the character of the method when concurrent steps are executed.

The method in question was developed in 1990 by Cook, Rutherford, Scarf and Shallcross. To date, we have been able to develop a sequential implementation of their algorithm that is capable of solving some small but very difficult mixed integer programs that arise in the design of fiber optic networks. These problems are intractable with standard Branch and Bound approaches. Our sequential code has been tested on the Intel 1860 hypercube, and the preliminary steps needed to obtain a parallel implementation are nearly complete.

The algorithm that we have implemented proceeds by creating a large search tree, and using a very complicated routine to determine which branch to follow. Our focus thus far has been on parallelizing the branching step of the overall process, and it is our belief that this will drastically reduce the time required for solving a problem. Once this step has been completed, we plan to investigate possible parallelization of the complicated routine that controls the branching sequence. This routine is invoked within every node of the large search tree that is generated by the algorithm, and we hope that by finding a viable parallelization of this routine, the running time of our code will be reduced enough that we will be able to solve problems with a larger number of integer variables.

Polyhedral Approaches

This work focuses on the study of the underlying polyhedral structure and the development of a branch-and-cut IP solver for a class of structured 0/1 integer programs arising from a truck dispatching scheduling problem. The problem is characterized by a group of set partitioning constraints and a group of knapsack equality constraints of a specific form. Families of facets for the polytopes associated with individual knapsack constraints are identified, and in some cases, a complete characterization of a polytope is obtained. In addition, a notion of "conflict graph" is introduced and utilized to obtain an approximating node-packing polytope for the convex hull of all 0/1 solutions. We are in the process of developing the branch-and-cut solver which generates cuts based on both the knapsack constraints and the approximating node-packing polytope, and incorporates these cuts into a tree-search algorithm that uses problem reformulation and LP-based heuristics at each node in

the search tree to assist in the solution process. Numerical experiments will then be performed on large-scale real instances supplied by an oil company in the United States. The optimal schedules obtained correspond to cost savings for the company and greater job satisfaction for drivers due to more balanced work schedules and income distribution. It is noteworthy that this is apparently the first time that branch-and-cut has been applied to an equality constrained problem in which the entries in the constraint matrix and right hand side are not purely 0/1.